

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matic Švab

**Razpoznavanje dreves s pomočjo
računalniškega vida**

DIPLOMSKO DELO
UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matej Kristan

Ljubljana 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Razpoznavanje dreves s pomočjo računalniškega vida

Tematika naloge:

Analizirajte problem razpoznavanja vrste dreves s pomočjo računalniškega vida na podlagi slike lubja. Preglejte literaturo s področja razpoznavanja tekstur in predlagajte primeren postopek predobdelave slike, luščenja značilnic in razpoznavanja. V kolikor primerna anotirana zbirka slik lubja ne obstaja, jo zgradite in anotirajte sami. Na zbirki analizirajte in kritično ovrednotite predlagani algoritem.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matic Švab, z vpisno številko **63090122**, sem avtor diplomskega dela z naslovom:

Razpoznavanje dreves s pomočjo računalniškega vida

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matjeja Kristana,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, 15. 9. 2014

Podpis avtorja:

Rad bi se zahvalil svoji družini za razumevanje, podporo in pomoč v času študija.

Iskreno se zahvaljujem svojemu mentorju doc. dr. Mateju Kristanu za njegovo pomoč in vodenje pri izdelavi diplomske naloge.

Prav tako bi se zahvalil vsem ostalim, ki so mi na kakršen koli način bili v pomoč in podporo v času študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Sorodna dela	3
1.3	Prispevki	5
1.4	Struktura	5
2	Teoretični okvir	7
2.1	Opisnik z lokalnim binarnim vzorcem	7
2.1.1	Osnovni LBP	7
2.1.2	Večnivojski LBP	10
2.1.3	Enakomeren LBP	11
2.1.4	Rotacijska robustnost	12
2.2	Metoda podpornih vektorjev	13
2.2.1	Linearni SVM	14
2.2.2	Mehka meja	17
2.2.3	Nelinearni SVM in večrazredna klasifikacija	17
2.2.4	Sferizacija	19
3	Predlagani postopki	21
3.1	Postopek razpoznavanja dreves	21

3.2	Postopek za samodejno določitev skale	23
4	Eksperimenti	25
4.1	Delovno okolje	25
4.1.1	MATLAB	25
4.1.2	Knjižnica VLFeat	27
4.2	Zbirka slik lubja	28
4.3	Izbira optimalne velikosti vhodnih slik	31
4.4	Izbira parametrov, jedra in metode za večrazredno klasificiranje	31
4.5	Preverjanje postopka za samodejno določanje skale	33
4.6	Preverjanje ugotovitev na celotni zbirki	36
5	Sklepne ugotovitve	39
5.1	Nadaljnje delo	40
	Slike	41
	Literatura	43

Seznam uporabljenih kratic

kratica	angleško	slovensko
LBP	Local binary patterns	Lokalni binarni vzorci
SVM	Support vector machine	Metoda podpornih vektorjev
SIFT	Scale-invariant feature transform	/
MATLAB	MATrix LABoratory	/
MEX	MATLAB Executable	/
RBF	Radial basis function	Radialna bazna funkcija
DoG	Difference of Gaussian	Razlika gaussov

Povzetek

Razpoznavanje objektov predstavlja obsežno področje v računalniškem vidu. V tej diplomski nalogi je predstavljen postopek razpoznavanja dreves s pomočjo računalniškega vida, ki analizira drevesno lubje. Iz posameznih slik lubja postopek najprej izlušči značilnice, ki jih SVM uporabi za učenje in testiranje. Za značilnice je uporabljen LBP, ki ima poleg hitrega izračuna še druge prednosti. Ker prosto dostopna zbirka slik drevesnega lubja ne obstaja, je bilo potrebno za izvedbo eksperimentov zajeti večjo anotirano zbirko slik lubja, ki je tudi prva javno dostopna zbirka. S podrobnejšo analizo oziroma z obsežnejšim eksperimentom so bili izbrani optimalni parametri, najboljše jedro in metoda za večrazredno klasifikacijo. Pri razpoznavanju se pojavi še problem določitev skale oziroma velikost slike, saj različne naprave zajamejo slike različnih velikosti, različnih razmerij širine/višine slike predvsem pa ljudje ne slikajo enako oddaljeni od dreves. V diplomski nalogi je tudi predlagan postopek, ki s pomočjo značilnic, pridobljenih s detektorjem DoG, samodejno določi skalo slike, s katero se vhodna slika pred izračunom LBP-ja vedno preskalira v referenčno velikost in s tem teži k normalizirani velikosti pomembnih struktur v slikah. Končni eksperiment je na zbirki 12 dreves dosegel 84.62 % natančnost.

Ključne besede: lokalni binarni vzorci, metoda podpornih vektorjev, klasifikacija dreves, samodejno določanje skale.

Abstract

The object recognition represents a large sub-domain in the computer vision. This thesis presents a process of a tree recognition by means of the computer vision, which analyses the tree bark. Firstly, the procedure extracts features from individual pictures of bark, which are later used for training and testing by SVM. LBP, which, in addition to its fast calculation, also has some other advantages, is used for these features. Since freely accessible collection of tree bark pictures does not exist, it was necessary to create a larger annotated collection which was used for experiments and is also the first database publicly available. The optimal parameters, the best core and method for multiclass classification were selected by a detailed analysis or more extensive experiments. In recognition there is also a problem with scale or picture size, because different devices take pictures of different sizes, in different width/height proportions and mostly people do not take photographs from the same distance. The thesis also proposes a procedure that by means of the features gained by DoG detector, automatically determines the picture scale, by means of which the input picture is always rescaled in the reference size before the calculation of LBP and so it tends to a normalized size of important structures in the pictures. In the final experiment the 84.62 % accuracy was achieved on the collection of 12 trees.

Keywords: local binary patterns, support vector machine, tree classification, automatic scale determination.

Poglavje 1

Uvod

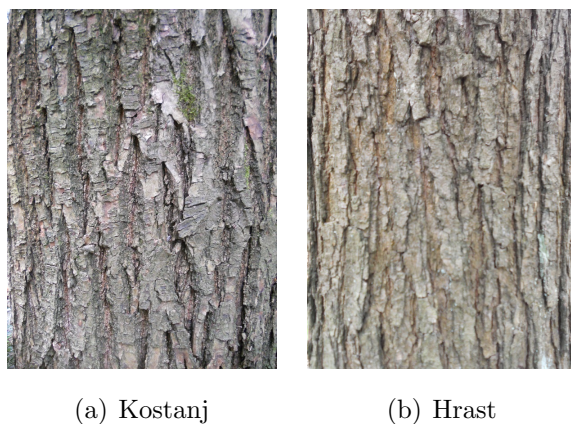
1.1 Motivacija

V vsakdanjem življenju se vedno več srečujemo z računalniškim vidom. Naj bo to od samodejnega krmiljenja industrijskih robotov, navigiranja avtonomnih vozil, sledenja ljudem s kamerami do človeško-računalniške interakcije (npr. Microsoftov Kinect) in različnih aplikacij na mobilnih telefonih, ki služijo za analiziranje slik. Vse te aplikacije uporabljajo računalniški vid, ki s pomočjo različnih metod za zajem, analiziranje in predelavo slikovnega materiala omogoča pridobitev numerične oziroma simbolične informacije. Pridobitve informacije lahko nato računalniški sistemi uporabijo pri svojih opravilih in odločitvah.

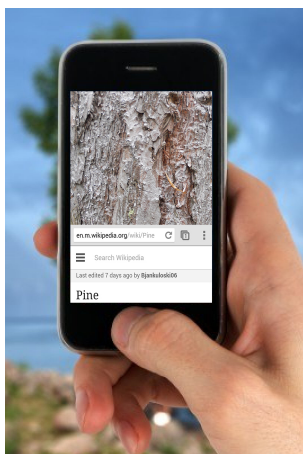
Eno izmed podpodročij računalniškega vida je razpoznavanje objektov. Smo v času hitrega razvoja mobilnih naprav; mobilni telefoni so v samo zadnjih nekaj letih dosegli velik napredek v strojni opremi oziroma zmogljivosti. Ta napredek omogoča izdelavo zahtevnejših mobilnih aplikacij, ki s pomočjo računalniškega vida razpoznavajo objekte.

V diplomski nalogi se bomo podrobneje posvetili problemu razpoznavanja dreves. Razpoznavanje dreves predstavlja zanimivo in hkrati tudi težavno nalogo. Če že samo pogledamo število vrst dreves, ki obstajajo, in pri tem upoštevamo še njihove podvrste, lahko hitro ugotovimo, da je ta številka

visoka. Poleg tega zagotovo želimo, da bi bilo razpoznavanje dreves možno skozi vse letne čase. Z listjem in plodovi si med časom zime ne moremo pomagati. Torej nam preostane samo še lubje, ki pa si je lahko med različnimi vrstami na videz podobno (npr. lubje kostanja je zelo podobno lubju hrasta, kar je tudi vidno na Sliki 1.1). Navsezadnje bi se po opravljeni raziskavi pridobljene ugotovitve lahko uporabile v potencialni mobilni aplikaciji, kot je vidna na Sliki 1.2.



Slika 1.1: Podobnost kostanja in hrasta.



Slika 1.2: Možnost uporabe postopka v aplikaciji.

1.2 Sorodna dela

Na področju samodejnega razpoznavanja dreves je bilo narejenih več raziskav. Chaki in Parekh [13] predlagata sistem za samodejno razpoznavanje rastlinja glede na njihovo obliko listov. Analize so narejene na treh različnih razredih rastlin s pomočjo dveh tehnik. Prva je model invariantnih trenutkov (ang. Moments-Invariant model), druga pa model radija ceentroida (ang. Centroid-Radii model). Z različnimi kombinacijami nastavitev za ta dva modela sta avtorja dosegla natančnost, ki se razteza med 90 % in 100 %. Predlagani postopek je dobro zasnovan, vendar pa ima dve opazni pomankljivosti. Pri segmentaciji mora biti list fotografiran na belem ozadju oziroma na listu papirja. Tu je tudi še zelo majhna množica razredov, ki zajema le 3 rastline.

Huang v svojem delu [18] predlaga postopek razpoznavanja lubja, ki temelji na osnovi barve in transformacije z valjčki (ang. wavelet texture filter). Ta postopek primerja še z drugimi značilnicami in metodami, kot so avto-korelacijska metoda, sopojavitvene matrike in metoda histograma. Za razpoznavanje uporablja dva klasifikatorja; to sta metoda podpornih vektorjev (ang. support vector machine, SVM) in RBPNN (ang. radial basis probabilistic network). Uporabil je lastno zbirko 300 slik lubja, za katero pa število razredov dreves ni bilo podano. Iz Tabele 1.1 je razvidno, da se je njegov predlagani postopek odrezal veliko bolje od ostalih.

značilnica/klasifikator	RBPNN	SVM
avto-korelacijska metoda	72 %	65 %
sopojavitvena matrika	77 %	75 %
metoda histograma	65 %	62 %
barva in transformacija z valjčki	84.68 %	82.26 %

Tabela 1.1: Primerjava natančnosti različnih značilnic v članku [18].

V članku [16] avtorja predlagata razpoznavanje dreves s pomočjo značilnic SIFT [20], ki jih pridobita iz slik lubja in listov (zbirko sta pridobila od Österreichische Bundesforste AG [5]). Problem SIFTa, ki ga avtorja tudi

omenita, je njegova računska zahtevnost. Za klasifikacijo je uporabljen SVM. V prvem poizkusu je bila na zbirki listov petih dreves dosežena 93.6 % natančnost. Sledil je drugi poizkus, kjer avtorja uporabita zbirko lubja enajstih dreves in pri tem dosežeta 69.7 % natančnost. Zbirko sta uporabila še pri preverjanju natančnosti razpoznavanja dveh strokovnjakov, kjer je prvi imel 56.6 % natančnost, drugi pa 77.8 %. Avtorja sta prav tako preverila možnosti za razpoznavanje iglavcev glede na iglice, vendar v svoji raziskavi nista našla nobenih dobrih možnosti za takšno razpoznavanje. Iz tega članka je razvidno, da je listje lahko dober izbor za klasifikacijo in tudi predlagana metoda za razpoznavanje po lubju se že lahko primerja z natančnostjo strokovnjakov.

Med samo izdelavo diplomske naloge smo zasledili še en zelo zanimiv članek [29]. Avtorja za razpoznavanje dreves predlagata postopek s SVM klasifikacijo več velikostnih opisnikov, narejenih iz sestavljenjih histogramov lokalnih binarnih vzorcev (ang. local binary patterns, LBP); s tem naj bi bila zagotovljena robustnost na velikostne spremembe. Za SVM je uporabljeno linearno jedro in več razredna metoda eden proti vsem. Postopek sta poizkusila na zbirki Österreichische Bundesforste AG (zbirka slik ni na voljo javnosti). Ker je ločljivost posameznih slik v zbirki različna, sta vsako sliko v zbirki, ki vsebuje 1182 slik enajstih različni razredov dreves, pred poizkusom spremenila na skupno velikost tisočih pikslov širine. Predlagani postopek je na omenjeni zbirki slik dosegel zelo visoko 96.5 % natančnost.

Omeniti je tudi vredno, da obstaja aplikacija za mobilne telefone, ki razpoznavo dreves opravi s slikanjem lista [19]. Prepozna lahko kar 184 različnih dreves v severo-vzhodnem delu ZDA. Postopek razpoznavanja je sestavljen iz štirih faz. Najprej se preveri, če slika vsebuje veljaven list za nadaljnjo uporabo, sledi segmentiranje, s katerim se pridobi binarno sliko, ki ločuje list od ozadja, iz binarne slike pa se nato iz večih velikosti pridobijo histogrami ukrivljenosti, iz katerih se sestavi HoCS (ang. Histogram of Curvature over Scale). Nazadnje sledi še primerjanje, ki vrne vrste dreves z najbližjim ujemanjem. Celoten postopek traja okrog 5 sekund. Aplikacija je zaenkrat na voljo samo iPhone in iPad napravam.

1.3 Prispevki

Diplomska naloga prinaša dva bistvena prispevka, glavni je postopek, ki s pomočjo opisnika LBP razpozna vrsto drevesa na podlagi lubja. Pri tem je tudi opravljena analiza, s katero so se določili optimalni parametri, najboljše jedro SVM-ja in najboljša metoda za večrazredno klasifikacijo. Posebnost izvedenega postopka je v tem, da naslavlja problem samodejnega določanja skale, s katero se vhodna slika pred izračunom LBP-ja preskalira v referenčno velikost.

Za potrebe diplomske naloge je bila potrebna zbirka slik lubja. Takšna javno dostopna zbirka ne obstaja, zato je drugi prispevek diplomske naloge novo zajeta zbirka slik lubja 12 različnih dreves, ki jih najdemo v Sloveniji. Ta zbirka je trenutno tudi prva največja javno dostopna zbirka.

1.4 Struktura

Preostanek diplomske naloge je sestavljen iz štirih glavnih poglavij. V Poglavju 2 sta podrobneje predstavljena opisnik z lokalnim binarnim vzorcem in metoda podpornih vektorjev kot klasifikator. V Poglavju 3 sledita natančna opisa postopka za razpoznavanje dreves in postopka za samodejno določanje skale. V Poglavju 4 so najprej predstavljeni delovno okolje in novo zajeta zbirka slik lubja. Sledi obsežnejši eksperiment, s katerim se določijo optimalni parametri, najboljše jedro in metoda za več razredno klasificiranje. V naslednjem eksperimentu je preverjen naš predlagan postopek za samodejno določanje skale in potreba po morebitnem filtriranju značilnic. Nazadnje so vse ugotovitve, uporabljene na celotni zbirki dreves. V zadnjem Poglavju 5 so povzete vse ugotovitve in predstavljene možnosti za nadaljnje delo.

Poglavje 2

Teoretični okvir

2.1 Opisnik z lokalnim binarnim vzorcem

Pri razpoznavanju dreves smo se odločili za opisnik z lokalnim binarnim vzorcem (ang. local binary pattern, LBP) [24], ki se je že v mnogih raziskavah izkazal za zelo uspešnega in ga je možno tudi zelo hitro izračunati, zato je tudi s stališča naše raziskave zelo primeren. V nadaljevanju bo zaradi lažjega razumevanja opisan najosnovnejši LBP, s katerim je opisana izbrana regija teksture. Sledile bodo še nekatere izmed razširitev LBP-ja, ki lahko izboljšajo natančnost klasificiranja.

2.1.1 Osnovni LBP

Leta 1996 je Ojala predlagal opisnik, ki ga je opisal v članku [24] in poimenoval kot lokalni binarni vzorec ali na kratko LBP (ang. local binary pattern). Zasnovan je bil na osnovi predpostavke, da ima tekstura lokalno dva komplementarna aspekta, to sta vzorec in njegova moč [27]. Zaradi svoje računske preprostosti, dobre zmogljivosti in odpornosti na enakomerne spremembe v osvetlitvi, je hitro postal eden izmed močnih opisnikov teksture.

Postopek osnovnega LBPja je preprost in se ga lahko opiše v naslednjih korakih [1] (za lažje razumevanje je postopek še prikazan na Sliki 2.1):

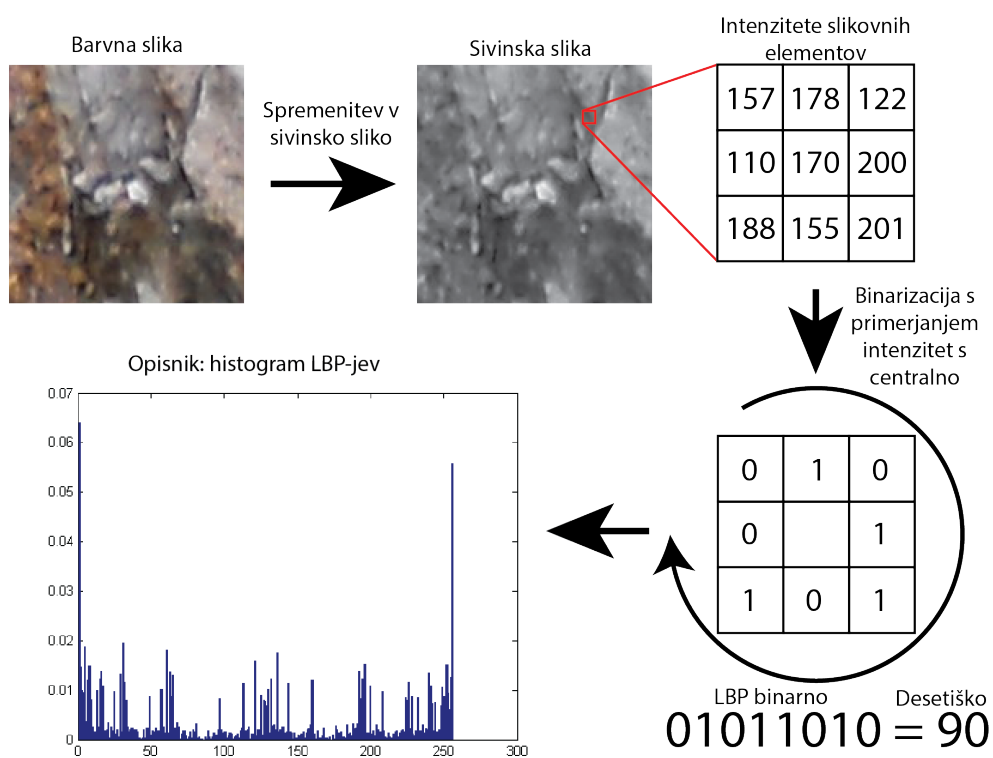
1. Sliko pretvorimo v sivinsko sliko.
2. Vsak piksel v sliki primerjamo z vsakim od osmih sosednjih pikslov.
3. Kjer je vrednost sredinskega piksla večja od sosednjega, se v sosednji piksel zapiše 0, sicer pa 1.
4. S tem dobimo 8-bitno binarno število, ki ga pretvorimo v desetiško število (okrog sredinskega piksla v smeri urinega kazalca).
5. Iz izračunanih LBP-jev vsakega piksla oziroma dobljenih desetiških števil tvorimo histogram.
6. Histogram še normaliziramo.

Histogram se lahko zapiše tudi z (2.1), pri kateri I in J predstavljata velikost slike, K pa največjo LBP vrednost v sliki.

Enačba LBP histograma:

$$H(k) = \sum_{i=1}^I \sum_{j=1}^J f(LBP(i, j), k), k \in [0, K] \quad (2.1)$$

$$f(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{sicer} \end{cases}$$



Slika 2.1: Izračun značilnic LBP.

Za primerjavo histogramov lahko uporabimo standardne mere razdalj [27], ki so vidne v (2.2), (2.3), (2.4) in (2.5). V spodnjih enačbah S in M predstavljata histograma vzorca in modela, S_b in M_b pa posamezno celico v histogramu. B predstavlja število vseh celic v histogramu.

G statistika:

$$G(S, M) = 2 \sum_{b=1}^B [S_b \log S_b - S_b \log M_b] \quad (2.2)$$

Logaritemska verjetnostna statistika:

$$L(S, M) = - \sum_{b=1}^B S_b \log M_b \quad (2.3)$$

Razdalja hi-kvadrat:

$$\chi^2(S, M) = \sum_{b=1}^B \frac{(S_b - M_b)^2}{S_b + M_b} \quad (2.4)$$

Presek histogramov:

$$H(S, M) = \sum_{b=1}^B \min(S_b, M_b) \quad (2.5)$$

2.1.2 Večnivojski LBP

Ena izmed pomanjkljivosti osnovnega LBP-ja je, da značilnice, izračunane v majhnem 3×3 velikem področju, ne morejo zajeti velikih struktur, ki so v nekaterih teksturah lahko prevladajoče [27]. Zato so nekateri avtorji predlagali, da se LBP izračuna tudi s pomočjo bolj oddaljenih sosednjih pikslov. S tem je LBP pridobil dva parametra, od katerih P pomeni število uporabljenih sosednjih pikslov in R , ki predstavlja polmer krožnice, na kateri se nahajajo sosednji piksli. Ta primer LBP-ja kot tudi osnovni LBP (za njega se uporabita parametra $P = 8$ in $R = 1$) se lahko izračuna po (2.6) [17], pri kateri so še x_c in y_c koordinate piksla, g_c in g_p pa sivinski vrednosti središčnega piksla in posameznih sosednjih pikslov. Koordinate sosednjih pikslov dobimo s $(x_p, y_p) = (R \cos(\frac{2\pi p}{P}), R \sin(\frac{2\pi p}{P}))$. Vrednosti sosednjih pikslov, ki ne padejo v središča pikslov, se izračunajo s bilinearno interpolacijo bližnjih pikslov.

Enačba LBP za posamezen piksel:

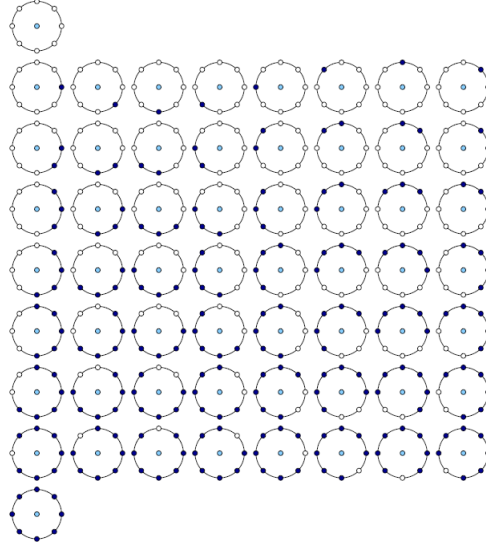
$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.6)$$

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & \text{sicer} \end{cases}$$

2.1.3 Enakomeren LBP

Še ena izmed razširitev osnovnega LBP-ja so enakomerni vzorci, ki jih Ojala in sod. opišejo v članku [25]. Pri njih se uvede vrednost U , ki pomeni število bitnih sprememb iz 0 v 1 ali obratno, pri čemer je vzorec gledan kot ciklični. LBP je enakomeren, če vsebuje največ dve takšni spremembi [8, 27], npr. vzorci 00000000 (0 sprememb), 11001111 (2 spremembi) in 01110000 (2 spremembi) so enakomerni vzorci, medtem ko 11001001 (4 spremembe) in 01010011 (6 sprememb) nista enakomerna vzorca. S to uvedbo je vsak enakomeren vzorec preslikan v svojo izhodno vrednost, vsi neenakomerni pa v eno samo vrednost. Pri parametrih ($P = 8, R = 1$) je to 58 enakomernih vzorcev, ki jih lahko vidimo na Sliki 2.2. Pri tem se v LBP histogramu število celic zmanjša s 256 na 59 celic oziroma gledano splošno, je število celic potem enako $P(P-1)+3$, kjer je P število sosednjih pikslov.

Razloga za izpust neenakomernih vzorcev sta dva. Ojala in sod. [25] so v svojih poizkusih s teksturami opazili, da je enakomernih vzorcev malo manj kot 90 % z uporabo ($P = 8, R = 1$) in okrog 70 % z uporabo ($P = 16, R = 2$). Podoben poizkus je bil narejen tudi na slikah obrazov [9], kjer je bilo enakomernih 90.6 % vzorcev pri ($P = 8, R = 1$) in 85.2 % pri ($P = 8, R = 2$). Drugi razlog za izpust neenakomernih vzorcev je statistična robustnost. Enakomerni vzorci so po nekaterih ugotovitvah bolj stabilni oziroma bolj odporni na šum. Namesto uporabe vseh možnih vzorcev uporaba enakomernih vzorcev ponekod omogoča boljše razpoznavne rezultate v različnih aplikacijah.



Slika 2.2: Prikaz 58 različnih kombinacij enakomernih vzorcev.

2.1.4 Rotacijska robustnost

V slikah se lahko zgodi, da so iste teksture rotirane v neko smer. To pomeni, da za vsako različno rotacijo dobimo različno LBP vrednost za posamezen piksel, kar pomeni, da ob različnih rotacijah dobimo tudi različen LBP histogram. S preprostim načinom lahko dosežemo rotacijsko robustnost. Vsako binarno LBP kodo ciklično rotiramo v njeno najmanjšo binarno vrednost. To lahko prikažemo z enačbo $LBP_{P,R}^{ri} = \min_i ROR(LBP_{P,R}, i)$, kjer $ROR(x, i)$ pomeni desno ciklično rotacijo bitov v zaporedju x za i korakov, npr. 8 bitne LBP kode 10000010, 00101000 in 00000101 se spremenijo v isto minimalno kodo 00000101. Seveda poleg tega preprostega načina obstajajo še drugi kompleksnejši, kot je način opisan v delu [27], kjer se rotacijska robustnost doseže s transformacijo histogramov.

2.2 Metoda podpornih vektorjev

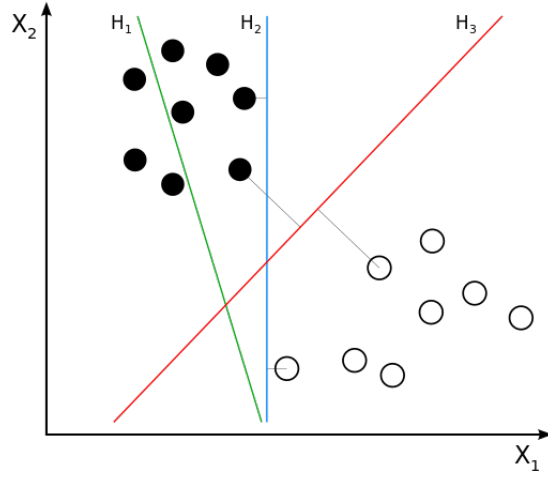
V statistiki in strojnemu učenju klasifikacija pomeni določitev, kateri kategoriji pripada nek nov podatek. Temu se kategorija določi na podlagi učne množice, v kateri je za vsak podatek, ki je v množici, kategorija znana. Podatki so analizirani z merljivimi lastnostmi, ki so lahko kategorične (npr. A, B, AB ali 0, za primer krvne skupine), vrstilne (majhno, srednje, veliko), numerične (npr. število pojavitev besed) in realne vrednosti (npr. meritev krvnega pritiska) [6]. Algoritem, ki implementira klasifikacijo, se imenuje klasifikator. V strojnem učenju so podatki pogosto imenovani kot primer in lastnosti kot značilnice, ki so združene v vektor značilnic, kategorije pa imenujemo razredi.

Obstaja veliko različnih klasifikatorjev; vsak ima svoje prednosti in slabosti. Najbolj pogosto uporabljeni klasifikatorji so nevronska mreža [23], K-najbližjih sosedov [10], odločitveno drevo [28] in SVM (ang. support vector machines) [7]. Uspešnost razvrščanja je v veliki meri odvisna od značilnosti podatkov, ki se klasificirajo. SVM je najbolj razširjen in v primerjavi z ostalimi klasifikatorji dosega zelo dobre rezultate, zato smo ga uporabili v naših eksperimentih in bo v nadaljevanju tudi podrobneje predstavljen.

SVM je način klasificiranja primerov pri strojnem učenju, ki je v pogosti rabi. V osnovi gre za binarno linearni klasifikator, ki podatke klasificira v enega izmed dveh razredov. Seveda SVM ni omejen samo na linearno klasifikacijo in dva razreda. S t. i. trikom jeder (ang. kernel trick) lahko izvedemo tudi nelinearno klasifikacijo in jo z malo logike razširimo na večrazredno. Obe razširitvi bosta opisani v nadaljevanju podpoglavja.

Osnovno idejo sta leta 1963 izdelala Vapnik in Lerner [30]. Vapnik je kasneje tudi poskrbel za različne izboljšave. Tekom let se je SVM spremenjal in različica, kot jo poznamo sedaj, je največ bila razvita v AT&T Bell Laboratories in nato leta 1995 tudi objavljena [15].

Kot rečeno SVM v osnovi klasificira med dvema razredoma. Da pa bomo lahko spoznali osnovno delovanje, domnevajmo, da imamo podatkovne točke,



Slika 2.3: Možnosti različnih hiperravnin.

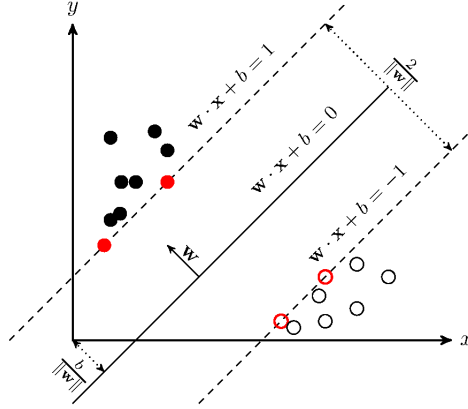
ki pripadajo enemu izmed dveh razredov. Cilj je seveda določiti razred novim podatkovnim točkam. Na te lahko gledamo kot p -dimenzionalni vektorji, ki jih želimo glede na njihova razreda ločiti s $(p-1)$ dimenzionalno hiperravnino. Kot je vidno na Sliki 2.3, lahko obstaja več različnih hiperravnin, vendar v osnovi želimo takšno, ki bo imela največjo ločevalno mejo med obema razredoma (na Sliki 2.3 je to hiperravnina H_3). Rečeno drugače, izberemo tisto hiperravnino, ki ima maksimirano razdaljo med hiperravnino in najbližjima podatkovnima točkama na obeh straneh hiperravnine. Takšno hiperravnino poimenujemo optimalna (ang. maximum-margin hyperplane). Na podlagi te lahko nato novim podatkovnim točkam, glede na katero stran padejo, določimo njihov razred.

2.2.1 Linearni SVM

Imamo množico učnih podatkov, ki jih sestavlja n točk oblike:

$$D = \{(\mathbf{x}_i, y_i) | \mathbf{x}_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n \quad (2.7)$$

Pri tem \mathbf{x}_i predstavlja točko, ki je p -dimenzionalen realni vektor [7]. Vsaki točki pripada tudi y_i , ki je lahko 1 ali -1 in predstavlja razred, ki mu točka



Slika 2.4: Optimalna hiperravnina.

\mathbf{x}_i pripada. Predpostavi se tudi, da so točke linearno ločljive, kar pomeni, da lahko določimo hiperravnino, ki večdimenzionalen prostor razdeli na dva dela. Takšno hiperravnino lahko zapišemo z (2.8), kjer \mathbf{w} predstavlja utežni vektor in b , ki predstavlja nagnjenost (ang. bias).

$$\mathbf{w} \cdot \mathbf{x}_i + b = 0 \quad (2.8)$$

Na podlagi predpostavke, da so točke linearno ločljive, obstaja takšen par (\mathbf{w}, b) , da za točke velja [12]

$$\begin{aligned} \mathbf{w} \cdot \mathbf{x}_i + b &\geq 1, y_i = 1, \\ \mathbf{w} \cdot \mathbf{x}_i + b &\leq -1, y_i = -1, \end{aligned} \quad (2.9)$$

skupaj z odločitvenim pravilom,

$$f_{\mathbf{w},b}(\mathbf{x}) = \text{sgn}(\mathbf{w} \cdot \mathbf{x} + b). \quad (2.10)$$

Enačbi (2.9) lahko tudi združimo v eno samo

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, 1 \leq i \leq n. \quad (2.11)$$

Mejo, ki ločuje razreda (vidno na Sliki 2.4), z dolžino $\frac{2}{\|\mathbf{w}\|}$, želimo maksimirati oziroma je enako, če $\|\mathbf{w}\|$ minimiziramo. Ker je predstavljeni optimizacijski problem težavno rešiti (zaradi odvisnosti od $\|\mathbf{w}\|$), lahko $\|\mathbf{w}\|$

nadomestimo z $\frac{1}{2}\|\mathbf{w}\|^2$. S tem dobimo kvadratični programski optimizacijski problem[12]

$$\begin{aligned} \text{Minimize } \Phi(\mathbf{w}) &= \frac{1}{2}\|\mathbf{w}\|^2, \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1, \quad i = 1, \dots, n. \end{aligned} \quad (2.12)$$

Optimizacijski problem (2.12) lahko rešimo s pomočjo Lagrangeovega multiplikatorja, ki je za dani problem enak

$$L(\mathbf{w}, b, \Lambda) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \lambda_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1], \quad (2.13)$$

kjer so $\Lambda = (\lambda_1, \dots, \lambda_n)$ Langrangeovi multiplikatorji za vsako točko. Problem rešimo z maksimiranjem L glede na $\Lambda \geq 0$ in minimiziranjem glede na \mathbf{w}, b . Z odvajanjem po \mathbf{w} in b ter postavitvijo odvoda na 0 prinese naslednji dve enačbi

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^n \lambda_i y_i \mathbf{x}_i = 0, \\ \frac{\partial L(\mathbf{w}, b, \Lambda)}{\partial b} &= - \sum_{i=1}^n \lambda_i y_i = 0. \end{aligned} \quad (2.14)$$

Optimalna rešitev je torej dana z utežnim vektorjem

$$\mathbf{w} = \sum_{i=1}^n \lambda_i y_i \mathbf{x}_i. \quad (2.15)$$

Z vstavitvijo enačb (2.14) in (2.15) v enačbo (2.13) dobimo [12],

$$F(\Lambda) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \quad (2.16)$$

ki vodi v dualni problem

$$\begin{aligned} \text{Maximize } &F(\Lambda), \\ \Lambda &\geq 0, \sum_{i=1}^n \lambda_i y_i = 0. \end{aligned} \quad (2.17)$$

Upoštevati je potrebno, da so Lagrangeovi multiplikatorji neničelni, kadar je $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1$, vektorji, na katere se to nanaša, pa imenujemo podporni (ležijo najbližje ločitveni hiperravnini). Optimalne uteži dobimo z enačbo (2.15), optimalen b pa z enačbo

$$b = y_i - \mathbf{w} \cdot \mathbf{x}_i. \quad (2.18)$$

Nato določitveno pravilo zapišemo kot

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^n y_i \lambda_i \mathbf{x} \cdot \mathbf{x}_i + b \right). \quad (2.19)$$

2.2.2 Mehka meja

Cortes in Vapnik [15] sta leta 1995 predlagala spremenjeno mejo, ki dovoljuje napačno klasificirane primere. Mehka meja v primeru, da hiperravnina, ki bi ločila primere, ne obstaja, izbere hiperravnino, ki loči primere, kar se da čisto. Hkrati še vedno maksimira razdaljo do najbližje čiste razdelitve primerov [7]. Uvedemo zrahljane spremenljivke (ang. slack variables), ki merijo stopnjo napačne klasifikacije podatkov. Optimizacijski problem (2.12) nato postane

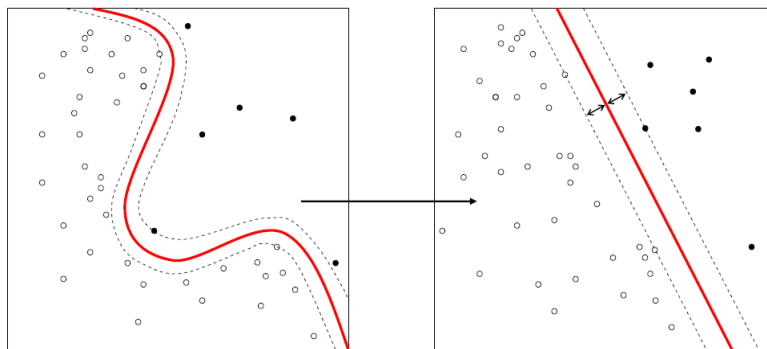
$$\begin{aligned} \text{Minimize } \Phi(\mathbf{w}) &= \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^n \xi_i, \\ y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \dots, n. \end{aligned} \quad (2.20)$$

C je prosti parameter, ki nadzira relativno pomembnost zmanjševanja norme \mathbf{w} in zadovolji omejitveno mejo vsake podaktovne točke [26]. Tudi tu lahko problem (2.20) rešimo s pomočjo Lagrangeovega multiplikatorja. Želimo torej maksimirati $F(\Lambda)$

$$\begin{aligned} F(\Lambda) &= \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j, \\ 0 \leq \Lambda \leq C, \quad \sum_{i=1}^n \lambda_i y_i &= 0. \end{aligned} \quad (2.21)$$

2.2.3 Nelinearni SVM in večrazredna klasifikacija

Osnovi SVM, ki sta ga Vapnik in Lerner predlagala leta 1963, je linearni klasifikator. V primeru, da učni podatki D niso linearni, ne moremo določiti hiperravnine, ki bi podatke ločila. Zato je bilo leta 1992 predlagano, da se ustvari nelinearne klasifikatorje s trikom z jedri (ang. kernel trick) [7]. SVM je še vedno podoben osnovnemu, le da se vsak skalarni produkt zamenja z nelinearno jedrno funkcijo. S tem lahko dobimo optimalno hiperravnino v transformiranem prostoru značilnic, kot je vidno na Sliki 2.5.



Slika 2.5: Transformacija nelinearnega prostora v visoko dimenzijski prostor.

Nekateri primeri pogosto uporabljenih jeder so:

Polinomsko jedro

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^d, \quad (2.22)$$

Gaussovo jedro

$$k(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \quad (2.23)$$

Gaussova radialna bazna funkcija (RBF)

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma \geq 0, \quad (2.24)$$

Laplaceova radialna bazna funkcija (RBF)

$$k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|}, \gamma \geq 0. \quad (2.25)$$

Do sedaj smo predpostavili, da sta samo dva razreda pri klasifikaciji, vendar imamo navadno opravka z večjim številom razredov. Binarni SVM, kot ga poznamo, ne zadostuje in ga je potrebno razširiti oziroma moramo uporabiti drugačen pristop. Najbolj pogost način je, da en sam večrazredni problem razbijemo na več binarnih klasifikacij.

Dva takšna načina sta:

- EDEN PROTI VSEM (ang. one against all)

Ta način ima N različnih binarnih klasifikatorjev. Za i -ti klasifikator so pozitivni primeri tisti, ki so v razredu i , negativni primeri pa vsi ostali, ki niso v razredu i . Razred se določi s strategijo, zmagovalec vzame vse (klasifikator z največjo izhodno funkcijo).

- EDEN PROTI ENEMU (ang. one against one)

Ta način ima $\frac{N(N-1)}{2}$ različnih binarnih klasifikatorjev, saj primerjamo vse možne pare razredov. Klasifikacija se opravi s pomočjo glasovanja. Vsak klasifikator določi podatek enemu izmed dveh razredov, s tem se tudi določenemu razredu glas poveča za ena. Končni razred se nato določi s razredom, ki ima največ glasov.

2.2.4 Sferizacija

Natančnost klasificiranja se lahko v nekaterih primerih izboljša z uporabo t. i. sferizacije, kjer se podatki centrirajo s pomočjo formule

$$\hat{x}_i = \frac{x_i - \mu_{train}}{\sigma_{train}}, \quad (2.26)$$

kjer x_i predstavlja vektor podatkov, μ_{train} povprečno vrednost vseh podatkov v učni množici, σ_{train} pa standardni odklon. Omenjena skalarja se uporabi na vsakem elementu x_{ij} , ki predstavlja j -ti element v vektorju x_i . Formulo se uporabi tako na učni kot na testni množici, vendar se morajo pri testni množici za μ_{train} in σ_{train} uporabiti vrednosti izračunani iz učne množice.

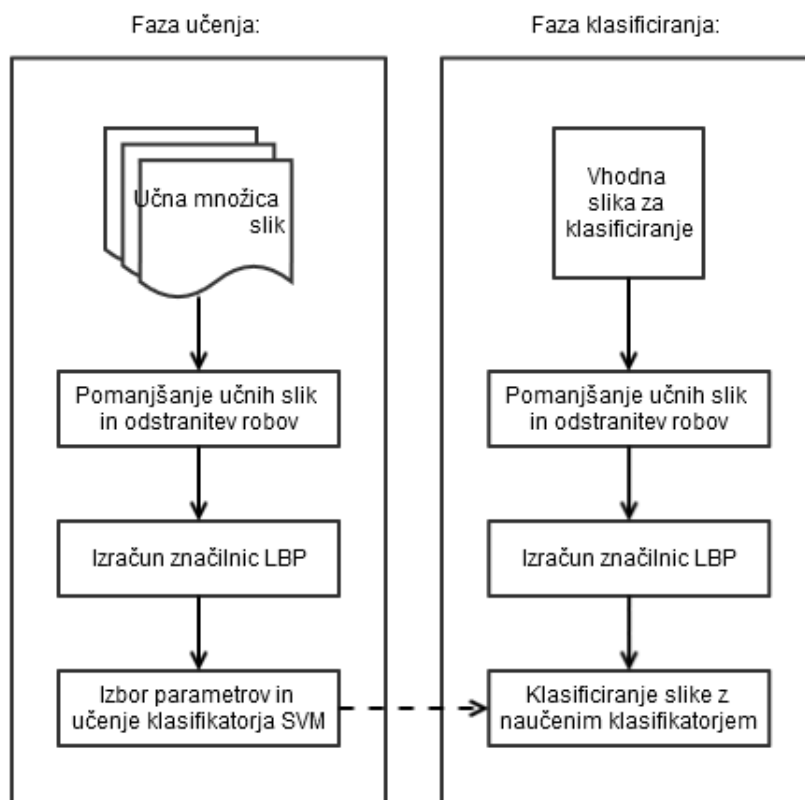
Poglavje 3

Predlagani postopki

3.1 Postopek razpoznavanja dreves

V tem podpoglavju bo predstavljena osnova za implementacijo postopka za razpoznavanje dreves, ki se ji lahko za lastne potrebe zelo enostavno dodajo nove funkcionalnosti. Za lažje razumevanje je postopek razdeljen na več ključnih korakov.

1. Najprej se določi učna množica slik, ki se bo uporabila pri učenju SVMja.
2. Učne slike se po potrebi pomanjša na ustrezno velikost (pri prvotni velikosti določeni postopki, kot npr. izračun LBP-ja, porabi veliko več časa kot pri manjših slikah, prav tako je lahko natančnost klasificiranja pri prvotnih slikah slabša kot pri manjših) in nato poreže robove, s čimer se odpravi morebitno šumno ozadje.
3. Za vsako učno sliko se izračunajo značilnice LBP, kot je opisano v Poglavju 2.1.1, in shrani njen razred.
4. SVM-ju se nastavijo parametri (C in pri RBF jedru še g) ter izbere željeno jedro in metodo za večrazredno klasifikacijo.



Slika 3.1: Diagram klasificiranja.

5. Značilnice LBP učnih slik in njihove znane razrede se uporabi za učenje SVM klasifikatorja.
6. Tudi sliko, ki se bo klasificirala, se po potrebi pomanjša, poreže robove in iz nje izračuna značilnice LBP.
7. Značilnice LBP se uporabi pri klasificiranju slike s klasifikatorjem SVM opisanim v Poglavju 2.2.

3.2 Postopek za samodejno določitev skale

Postopek, omenjen v prejšnjem podpoglavju, je uporabil predpostavko, da so vse pomembne strukture v lubju v vseh slikah enake velikosti. V primeru uporabe postopka v mobilni aplikaciji se pojavi problem, da različni mobilni telefoni fotografirajo slike pri različnih velikostih, različnih razmerjih širine/višine slike in predvsem ljudje ne slikajo enako oddaljeni od dreves, zato tudi ne bi bilo vedno smiselno slik pomanjšati s fiksno skalo. V tem podpoglavju je opisan postopek za samodejno določanje skale, s katero se vhodna slika pred izračunom LBPja vedno preskalira v referenčno velikost in s tem teži k normalizirani velikosti pomembnih struktur v slikah.

Izhajamo iz predpostavke, da se nekakšno informacijo o skali lahko pridobi s slike oziroma bolj natančno gledano iz teksture lubja. Za pridobitev takšne informacije o skali se uporabijo značilnice, ki se pridobijo s pomočjo detektorja DoG (ang. Difference of Gaussian) [31]. Te značilnice imenujemo packe (ang. blobs), ki so detektirane na več skalah, vsaka detektirana packa pa ima tudi skalo. Predlagani postopek se glede na postopek opisanega v Poglavju 3.1 uporabi pred izračunom značilnic LBP klasificirane slike. Za lažje razumevanje je tudi tu postopek povzet v ključne korake.

1. Za vsako sliko v učni množici se detektira značilnice z detektorjem DoG (enako pomanjšane učne slike kot pri postopku v Poglavju 3.1).
2. Iz značilnic vsake učne slike se izračuna mediano njihovih velikosti (velikosti se po potrebi filtrira, s čimer se doseže veliko boljše rezultate).
3. Iz median se tvori histogram z 10 celicami (enakomerno široke celice med najmanjšo in največjo vrednostjo median) in nato izbere mediano M_{ref} , pri kateri histogram doseže vrh.
4. Tudi za sliko, ki se klasificira, se detektira značilnice in izračuna mediano njihovih velikosti, ki se označi kot M_{det} .
5. Skalo pomanjšanja slike se dobi po formuli $S = \frac{M_{ref}}{M_{det}}$.

Poglavje 4

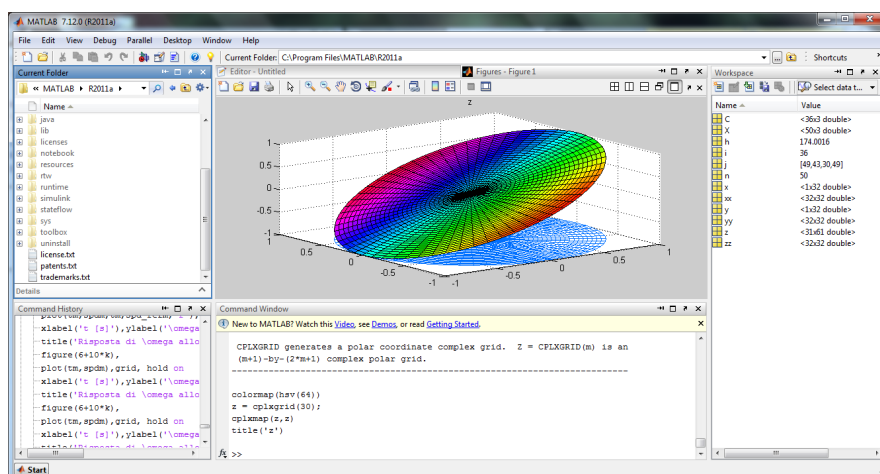
Eksperimenti

4.1 Delovno okolje

4.1.1 MATLAB

Programerski del diplomske naloge je bil narejen s pomočjo okolja MATLAB [4], ki poleg svoje velike zbirke funkcij nudi tudi dobro dokumentacijo o uporabi.

MATLAB je visokonivojski programski jezik četrte generacije in tudi interaktivno okolje (vidno na Sliki 4.1) za numerično računanje, simuliranje, programiranje, razvoj algoritmov, vizualizacijo in analizo podatkov [2]. Prvi koraki razvoja so se začeli že v poznih 70-tih prejšnjega stoletja, ko je Cleve Moler želel svojim študentom omogočiti dostop do LINPACK in EISPACK (programski knjižnici za linearno algebro) brez potrebe poznavanja Fortrana. Leta 1983 je med gostovanjem Molerja na Univerzi Stanford inženir Jack Little opazil tržni potencial v njegovi rešitvi, kar je vodilo v sodelovanje z Molerjem in Steveom Bangertom. Skupaj so leta 1984 ustanovili podjetje Mathworks, prepisali MATLAB v programski jezik C in nato nadaljevali z njegovim razvojem. Za industrijske in akademske namene ga uporablja že več kot milijon znanstvenikov in inženirjev.



Slika 4.1: Okolje MATLAB.

Glavne značilnosti MATLABa [3] so:

- interaktivno okolje za iterativno raziskovanje, oblikovanje in reševanje problemov
- visokonivojski jezik za numerično računanje, vizualizacijo in razvoj aplikacij
- vgrajena grafika za prikaz podatkov
- matematične funkcije za linearno algebro, statistiko, filtriranje, optimizacijo, Fourierjevo analizo ...
- orodja za izdelavo aplikacij s po meri narejenim grafičnim vmesnikom in orodja za izboljšanje kakovosti kode
- funkcije za povezovanje MATLABovih algoritmov z zunanjimi aplikacijami

MATLAB lahko uporabi funkcije, napisane v programskem jeziku C ali Fortranu. Da je takšna povezava možna, je potreben vmesnik med MATLAB-om in funkcijami. Za to povezavo poskrbi MEX datoteka, ki se po prevajanju

dinamično naloži in nato omogoča klic zunanje kode znotraj MATLAB-a, kot da bi bila vgrajena funkcija. Eden od primerov je v diplomski nalogi uporabljena knjižnica za SVM (LIBSVM) [14], ki je napisana v C-ju in omogoča obsežno nastavljivost parametrov. Poleg omenjene povezave se lahko v MATLAB-u direktno kličejo tudi knjižnice, napisane v Perlu, .Netu, ActiveXu in Javi.

Nekaj značilnosti sintakse:

- koda se lahko piše v ukazno okno, ki je vidno na Sliki 4.1, ali pa v datoteke s končnico `.m`, ki vsebuje skripte in funkcije
- MATLAB je šibko tipiziran programski jezik
- spremenljivke lahko ustvarimo brez določitve tipa
- indeksiranje matrik se prične z 1 in ne z 0 kot pri nekaterih drugih programskih jezikih
- vsebuje veliko operacij nad matrikami, ki odstranijo potrebo po počasnejših zankah
- ima razrede, katerih sintaksa in klic se razlikujeta od ostalih jezikov
- podpora za objektno-orientirano programiranje vsebuje dedovanje, razrede, podajanje po referenci, podajanje po vrednosti

4.1.2 Knjižnica VLFeat

Sedanje raziskave v računalniškem vidu so večinoma opravljene z različnimi algoritmi strojnega učenja, kot so pridobivanje značilnic, gručenje in učenje. Večina takšnih implementacij algoritmov je pogosto v binarni obliki in tudi za samo določene sisteme.

VLFeat [31] je odprto kodna knjižnica, ki implementira pogosto uporabljene algoritme v računalniškem vidu in omogoča hitro ustvarjanje prototipov in ponovljiv razvoj. VLFeat je napisan v programskem jeziku C, kar



Slika 4.2: Značilnice, pridobljene z detektorjem DoG.

omogoča učinkovitost in združljivost z MATLAB-om. Vsebuje veliko primerov uporabe in tudi dobro opisano kodo, kar omogoča, da jo uporabniki lahko preučujejo in spreminjajo [32]. Nekateri izmed primerov algoritmov, ki jih vsebuje, so detektor in opisnik značilnic SIFT [20], detektor značilnic MSER [22], naključna kd-drevesa [11], k-means [21], lokalni opisnik LIOP [34] in kovariantni detektorji značilnic, v katere spada tudi detektor DoG, ki smo ga v eksperimentih uporabili. Ta uporablja lokalno ekstremno sled večnivojskega Laplacejevega operatorja za detektiranje značilnic v skali in prostoru. Te značilnice, imenovane tudi kot packe, so orientirane elipse, vendar pa bodo za naše potrebe zadostovali neorientirani krogi (preprostejša oblika). Primer takšnih značilnic lahko vidimo na Sliki 4.2.

4.2 Zbirka slik lubja

Ker javno dostopna zbirka slik lubja ne obstaja, je bil eden izmed ciljev diplomske naloge ustvariti prvo javno dostopno zbirko, na kateri bomo lahko analizirali in ovrednotili predlagani postopek razpoznavanja dreves. S tem smo omogočili, da bodo drugi uporabniki lahko v nadaljnjih raziskavah uporabili že obstoječo zbirko in tudi nato lažje primerjali rezultate z drugimi

Lasnosti	
Število slikovnih pik	12MP
Slikovni senzor	1/2.3 palični CCD
Objektiv (zoom, goriščna razdalja, število f/)	4x, 4.9-19.6mm, f/3.2-5.9
Tip zajete slike	JPEG
Velikost slike	4000x3000

Tabela 4.1: Lasnosti fotoaparata.

uporabniki.

Zbirka slik vsebuje lubje 12 različnih dreves (bor, breza, bukev, gaber, hrast, jelša, kostanj, smreka, lipa, ginko, platana in divji kostan), ki jih najdemo v Sloveniji. Vse slike so bile fotografirane s fotoaparatom Nikon COOLPIX S3000; lastnosti so prikazane v Tabeli 4.1.

Pri fotografiranju so se upoštevali naslednji pogoji:

- enaka razdalja fotografiranja, ki znaša približno 20 cm
- primeri slik iste vrste, fotografirani na večjem številu dreves
- vsaj 30 različnih slik na eno vrsto drevesa
- izogibanje šuma na slikah (npr. lišaji na lubju, čim manj ozadja na slikah)
- fotografiranje pri približno enaki svetlosti
- slike so fotografirane v pokončni legi

Vsako drevo v zbirki vsebuje vsaj 30 različnih slik lubja, kar pomeni, da je v zbirki skupaj približno 360 slik. Primeri posameznih slik so vidni na sliki 4.3. Zbirka je tudi javno objavljena, in sicer na spletni strani laboratorija VICOS [33].



Slika 4.3: Primeri slik lubja v zbirki.

4.3 Izbira optimalne velikosti vhodnih slik

Najprej je bilo potrebno vse slike pomanjšati na primerno velikost, saj izračun značilnic LBP pri prvotni velikosti 3000 x 4000 pikslov porabi veliko časa. Prav tako velikost ni smela biti premajhna, saj s tem izgubimo preveč informacije o teksturi lubja in posledično to tudi povzroči slabše rezultate. Potrebno je bilo torej najti optimalno velikost, ki bo dala dobre rezultate, hkrati pa tudi ne bo porabila preveč časa pri izračunu značilnic LBP.

Iskanje optimalne velikosti smo se lotili tako, da smo za učno množico naključno izbrali 25 slik na vsako drevo, za učno množico pa 5. Pri tem smo uporabili le 8 dreves iz zbirke. Za učenje SVM-ja smo uporabili privzeto jedro oziroma linearno jedro. Nato smo za učne in testne slike izbrali začetno velikost, ki je bila 40 % prvotne velikosti in opravili 10 ponovitev klasificiranja z različnimi učnimi in testnimi množicami. V nadaljevanju smo slike pomanjševali po dodatnih 5 %, vse do 5 % prvotne velikosti (preverjali smo torej na intervalu 40–5 % prvotne velikosti) in ponovili klasificiranja. Na podlagi uspešnosti klasificiranja in upoštevanju časa izračuna značilnic LBP, smo se odločili, da je slike najboljše pomanjšati na 10 % prvotne velikosti. Pri tej velikosti se za izračun značilnic LBP porabi 2.2 sekundi. S tem smo torej dobili slike velikosti 300 x 400 pikslov oziroma 270 x 360 pikslov, saj slikam zaradi možnosti prisotnosti ozadja porežemo še 10 % robova.

4.4 Izbira parametrov, jedra in metode za večrazredno klasificiranje

Pri prvem obsežnem eksperimentu so nas zanimala tri vprašanja:

- Katero izmed dveh izbranih jeder SVM-ja (linearno ali RBF) je bolje uporabiti?
- Kakšni so dobri parametri za izbrano jedro?
- Katera večrazredna metoda daje boljše rezultate?

	Linearno	RBF
Eden proti enemu	C=3450 natančnost=83.26 %	C=2060, g=75.5 natančnost=89.63 %
Eden proti vsem	C=1970 natančnost=86.78 %	C=390, g=38 natančnost=91.30 %

Tabela 4.2: Natančnosti in optimalni parametri jedra brez uporabe sferizacije.

Ker smo v eksperimentu uporabili slike z enakimi velikostmi (10 % prvotne velikosti, kot smo ugotovili v Poglavju 4.3), smo za vse lahko značilnice LBP izračunali vnaprej. Z vnaprejšnjim izračunom značilnic LBP smo pohitrili izvajanje eksperimenta, saj smo se s tem izognili nepotrebnemu izračunu istih značilnic pri različnih učnih in testnih množicah. Pri eksperimentu smo vse primerjave opravili na 8 razredih dreves iz zbirke.

V prvi primerjavi smo primerjali, kako se linearno in RBF jedro obneseta pri večrazredni metodi eden proti enemu (ang. one against one), istočasno smo za obe jedri iskali tudi optimalne parametre. Da smo dobili dobro povprečno natančnost klasifikacije, smo testno klasificiranje ponovili 200-krat, pri vsaki ponovitvi smo seveda izbrali drugačno učno in testno množico (25 učnih in 5 testnih slik na vsako drevo), da smo se tako izognili prekomernemu prilaganju na podatke (ang. overfitting). Za iskanje optimalnih parametrov smo uporabili postopno povečanje parametrov, in sicer smo parameter C povečevali za 10, dodatni parameter g pri jedru RBF pa za 0.5 (oba parametra smo ločeno drug od drugega povečali 360-krat). V drugi primerjavi smo za večrazredno klasifikacijo namesto metode eden proti enemu uporabili metodo eden proti vsem (ang. one against all). V Tabeli 4.2 lahko opazimo, da se jedro RBF obnese bolje kot linearno. Pri večrazrednih metodah je edina večja opazna razlika pri linearnem jedru, kjer je metoda eden proti vsem imela za približno 3.5 % boljšo natančnost.

	Linearno	RBF
Eden proti enemu	C=1620 natančnost=89.83 %	C=1910, g=0.5 natančnost=89.64 %
Eden proti vsem	C=2270 natančnost=91.08 %	C=880, g=1 natančnost=89.99 %

Tabela 4.3: Natančnosti in optimalni parametri jedra z uporabo sferizacije.

Obe primerjavi smo ponovili, le da smo tokrat natančnost želeli izboljšati s sferizacijo. V Tabeli 4.3 opazimo, da sferizacija pri jedru RBF ni imela opaznega vpliva, pri linearnem pa lahko v obeh primerih opazimo izboljšanje v natančnosti.

Na podlagi pridobljenih ugotovitev smo se odločili, da bomo v nadaljnjih eksperimentih uporabili SVM klasifikator z jedrom RBF ($C=390$, $g=38$) in večrazredno metodo eden proti vsem.

4.5 Preverjanje postopka za samodejno določanje skale

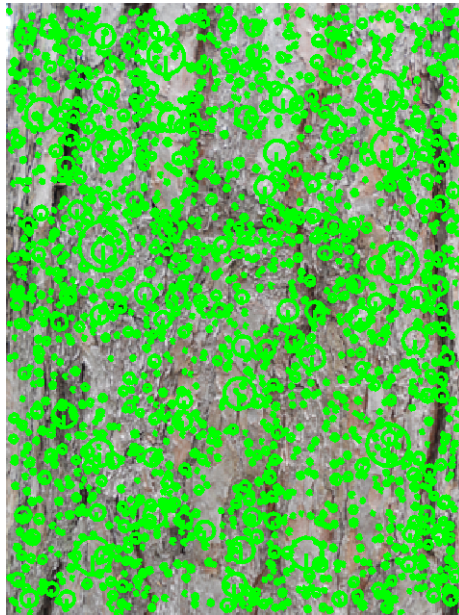
V tem eksperimentu nas je zanimalo, kako se obnese naš predlagani postopek za samodejno določanje skale, ki smo ga opisali v Poglavju 3.2 in če je potrebno značilnice, ki smo jih detektirali z detektorjem DoG, pred njihovo uporabo filtrirati. Tudi v tem eksperimentu smo uporabili 8 razredov dreves iz zbirke. Za učno množico slik smo lahko uporabili vnaprej izračunane značilnice LBP in tudi značilnice, pridobljene z detektorjem DoG (učne slike pri 10 % velikosti prvotnih slik). Testnim slikam smo pred njihovo uporabo naključno spremenili velikost na 10–30 % prvotne velikosti (s tem smo simulirali različne velikosti). V primeru testnih slik vnaprejšnji izračun obeh značilnic tokrat ni bil možen in so se zato izračunale sproti. Najprej nas je zanimalo, kako se klasificiranje obnese brez uporabe postopka. Po 200 ponovitvah z različnimi učnimi in testnimi množicami (tudi tu smo uporabili 25

Brez skale	57.78 %
Vse značilnice	62.17 %
2 % značilnic	67.98 %
Prvih 100 značilnic	81.99 %

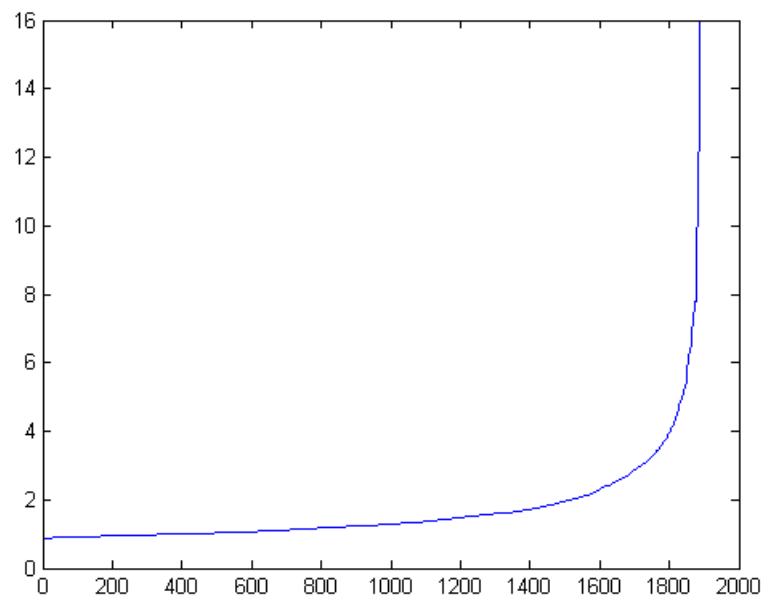
Tabela 4.4: Pridobljene natančnosti s posameznimi filtriranjem značilnic.

učnih in 5 testnih slik na drevo) smo dobili 57.78 % povprečno natančnost.

Nato smo uporabili naš predlagani postopek brez filtriranja značilnic. Za vsako testno sliko smo izračunali njeno skalo S in jo pred klasificiranjem uporabili. V tem primeru je povprečna natančnost sedaj znašala 62.17 %, kar je manj kot 5 % izboljšave. To ni bil ravno velik napredek, zato nas je zanimalo, če bo filtriranje detektiranih značilnic prineslo boljše rezultate. Na Sliki 4.4(a), ki prikazuje detektirane značilnice ene izmed slik v učni množici, je vidno, da vsebuje veliko podobno velikih drobnih značilnic. Te smo uredili po padajoči velikosti in nato za izračun skale poizkusili uporabiti le prvih 2 % največjih značilnic. V tem primeru se je natančnost povečala na 67.98 %, kar zopet ni bilo veliko. Odstotkovno filtriranje se je izkazalo za ne preveč učinkovito, saj se pri večjih slikah še vedno uporabi veliko drobnih značilnic. Na podlagi Slike 4.4(b) smo se odločili, da bomo značilnice filtrirali na prvih 100 največjih značilnic in te uporabili pri izračunu skale. To filtriranje se je izkazalo za bolj učinkovito, saj smo sedaj dosegli 81.99 % natančnost klasificiranja.



(a) Vse detektirane značilnice.



(b) Velikosti značilnic, urejene po velikosti, kjer y predstavlja velikost značilnic, x pa x -to značilnico.

Slika 4.4: Primer detektiranih značilnic učne slike.

4.6 Preverjanje ugotovitev na celotni zbirki

V vseh predhodnih eksperimentih so bile raziskave narejene na 8 drevesih, končna zbirka pa vsebuje slike 12 dreves. Razlog je, da se je zbirka med izvajanjem eksperimentov povečala za 4 dodatna drevesa. Zato smo se odločili, da končne ugotovitve in morebitne parametre, pridobljene iz prejšnjih poglavij, preverimo še na končni zbirki. Pri tem zadnjem eksperimentu so se torej upoštevale naslednje stvari:

- velikost učnih slik se je pomanjšala na 10 % prvotne velikosti
- za SVM klasifikator se je uporabilo RBF jedro ($C=390$, $g=38$) in večrazredna metoda klasificiranja eden proti vsem
- v postopku samodejnega določanje skale se je značilnice filtriralo na prvih 100 največjih značilnic

Tudi tu smo za dobro povprečno natančnost opravili 200 klasifikacij, vsakič z drugačno učno in testno množico. V primeru brez uporabe postopka za samodejno določanje skale smo dosegli 55.99 % natančnost, v primeru uporabe postopka pa visoko 84.62 %. Na Slikah 4.5 in 4.6 lahko tudi vidimo tipično matriko razvrstitev enega klasificiranja za omenjena primera.

Matrika razvrstitev

bor	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 5.0%	40.0% 60.0%
breza	0 0.0%	4 6.7%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
bukev	0 0.0%	0 0.0%	3 5.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	60.0% 40.0%
gaber	0 0.0%	3 5.0%	0 0.0%	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40.0% 60.0%
hrast	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 5.0%	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	60.0% 40.0%
jelša	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
kostanj	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 5.0%	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40.0% 60.0%
smreka	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	80.0% 20.0%
lipa	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	0 0.0%	1 1.7%	40.0% 60.0%
ginko	0 0.0%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	1 1.7%	20.0% 80.0%
platana	0 0.0%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 3.3%	1 1.7%	40.0% 60.0%
divji kostanj	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	80.0% 20.0%
													56.7% 43.3%
	bor	breza	bukev	gaber	hrast	jelša	kostanj	smreka	lipa	ginko	platana	divji kostanj	

Slika 4.5: Matrika razvrstitev brez samodejnega določanja skale.

Matrika razvrstitev

bor	5 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
breza	0 0.0%	5 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
bukev	0 0.0%	0 0.0%	1 1.7%	2 3.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 3.3%	0 0.0%	20.0% 80.0%
gaber	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
hrast	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
jelša	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
kostanj	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
smreka	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	0 0.0%	0 0.0%	0 0.0%	80.0% 20.0%
lipa	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	0 0.0%	100% 0.0%
ginko	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5 8.3%	0 0.0%	100% 0.0%
platana	0 0.0%	0 0.0%	0 0.0%	1 1.7%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	4 6.7%	80.0% 20.0%
divji kostanj	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 8.3%	100% 0.0%
												86.7% 13.3%
	bor	breza	bukev	gaber	hrast	jelša	kostanj	smreka	lipa	ginko	platana	divji kostanj

Slika 4.6: Matrika razvrstitev s samodejnim določanjem skale.

Poglavje 5

Sklepne ugotovitve

V diplomski nalogi smo obravnavali problem razpoznavanja dreves na podlagi lubja s pomočjo računalniškega vida. Predlagali smo postopek za razpoznavanje, ki za značilnice uporabi lokalne binarne vzorce, za klasificiranje pa metodo podpornih vektorjev. Kot izboljšavo pri razpoznavanju smo predlagali postopek za samodejno določanje skale. Tu smo uporabili značilnice, ki smo jih dobili s pomočjo detektorja DoG.

Najprej smo seveda potrebovali zbirko slik drevesnega lubja, kjer smo lahko nato izvajali naše eksperimente. Sprva je zbirka za eksperimentalne namene vsebovala 8 različnih dreves, kasneje pa je bila med izdelavo diplomske naloge razširjena na 12 dreves. Omenjena zbirka je tudi prva javno dostopna zbirka.

Pred glavnim eksperimentiranjem nas je tudi zanimalo, kakšna je optimalna velikost vhodnih slik, kjer dobimo dobre rezultate, istočasno pa čas izračuna ni prevelik. V kratkem eksperimentu smo se odločili za velikost, ki znaša 10 % prvotne velikosti slik.

V prvem večjem eksperimentu nas je najprej zanimalo, katero jedro (linearno ali RBF) in večrazredna metoda klasificiranja (eden proti enem ali eden proti vsem) pri uporabi SVM klasifikatorja daje najboljše rezultate. Istočasno smo iskali optimalen parameter C in za RBF jedro tudi parameter g . Po končanem eksperimentu so ugotovitve pokazale, da za dani problem

dobimo najboljše rezultate z uporabo RBF jedra in večrazredne metode eden proti vsem. Pri tem sta optimalna parametra $C=390$ in $g=38$.

V drugem eksperimentu smo preverili predlagani postopek za samodejno določanje skale in morebitno potrebo po filtriranju značilnic. Za primerjavo smo pred postopkom izvedli klasificiranje brez samodejnega določanja skale in pri tem dosegli natančnost 57.78 %. S klasificiranjem, kjer smo za izračun skale uporabili vse značilnice, smo s postopkom dosegli majhno izboljšavo, in sicer je natančnost znašala 62.17 %. Nato smo značilnice poskusili še filtrirati. Uredili smo jih po velikosti in vzeli prva 2 % največjih. Tudi v tem primeru smo dosegli majhno izboljšavo in smo se zato določili, da bomo vzeli le prvih 100 največjih. V tem primeru smo končno dosegli opazno izboljšavo v natančnosti, ki je sedaj znašala 81.99 %.

Nazadnje smo vse ugotovitve uporabili še v zadnjem eksperimentu, pri katerem smo namesto zbirke slik 8 dreves uporabili končno zbirko slik, ki je vsebovala 12 dreves. Pri tem eksperimentu smo dosegli presenetljivo 84.62 % natančnost.

5.1 Nadaljnje delo

Možnosti za nadaljnje delo je več. Kot najprej bi se lahko zbirka slik drevesnega lubja povečala, saj trenutnih 12 dreves v zbirki predstavlja zelo majhen delež med drevesi. Istočasno bi se še lahko naredila tudi zbirka slik listov dreves, saj se je v nekaterih že opravljenih raziskavah izkazalo, da kombinacija klasificiranja z lubjem in listi daje boljše rezultate. Izboljšave so možne tudi pri našem predlaganem postopku samodejnega določanja skale. Pri izbiri uporabnih značilnic za izračun skale bi se lahko namesto fiksnih 100 značilnic predlagal način, ki bi učinkovito določil mejo uporabnih značilnic. Seveda pa pri tem ne smemo pozabiti možnosti za nove predloge postopkov za samodejno določanje skale. Nazadnje bi se vse ugotovitve in postopek lahko uporabili v priročni aplikaciji za mobilne telefone.

Slike

1.1	Podobnost kostanja in hrasta.	2
1.2	Možnost uporabe postopka v aplikaciji.	2
2.1	Izračun značilnic LBP.	9
2.2	Prikaz 58 različnih kombinacij enakomernih vzorcev.	12
2.3	Možnosti različnih hiperravnin.	14
2.4	Optimalna hiperravnina.	15
2.5	Transformacija nelinearnega prostora v visoko dimenzijski prostor.	18
3.1	Diagram klasificiranja.	22
4.1	Okolje MATLAB.	26
4.2	Značilnice, pridobljene z detektorjem DoG.	28
4.3	Primeri slik lubja v zbirki.	30
4.4	Primer detektiranih značilnic učne slike.	35
4.5	Matrika razvrstitev brez samodejnega določanja skale.	37
4.6	Matrika razvrstitev s samodejnim določanjem skale.	38

Literatura

- [1] Local binary patterns. Dostopno na: http://en.wikipedia.org/wiki/Local_binary_patterns. Zadnji dostop: 30. 7. 2014.
- [2] Matlab. Dostopno na: <http://en.wikipedia.org/wiki/Matlab>. Zadnji dostop: 28. 7. 2014.
- [3] Matlab product description. Dostopno na: http://www.mathworks.com/help/matlab/learn_matlab/product-description.html. Zadnji dostop: 29. 7. 2014.
- [4] Matlab, version 8.1.0.604 (r2013a). The MathWorks Inc., Natick, Massachusetts.
- [5] Österreichische bundesforste ag. Dostopno na: <http://www.bundesforste.at/>. Zadnji dostop: 27. 8. 2014.
- [6] Statistical classification. Dostopno na: http://en.wikipedia.org/wiki/Statistical_classification. Zadnji dostop: 3. 8. 2014.
- [7] Support vector machine. Dostopno na: http://en.wikipedia.org/wiki/Support_vector_machine. Zadnji dostop: 3. 8. 2014.
- [8] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In *Computer Vision - ECCV 2004*, pages 469–481. Springer Berlin Heidelberg, 2004.

-
- [9] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.
 - [10] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
 - [11] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
 - [12] R. Burridge and B. Buxton. An introduction to support vector machines for data mining. Dostopno na: <http://svms.org/tutorials/BurridgeBuxton2001.pdf>. Zadnji dostop: 4. 8. 2014.
 - [13] J. Chaki and R. Parekh. Plant leaf recognition using shape based features and neural network classifiers. *International Journal of Advanced Computer Science and Applications*, 2(10):41–47, 2011.
 - [14] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011. Dostopno na <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. Zadnji dostop: 29. 7. 2014.
 - [15] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
 - [16] S. Fiel and R. Sablatnig. Automated identification of tree species from images of the bark, leaves and needles. *Proceedings of the 16th Computer Vision Winter Workshop*, pages 67–74, 2011.
 - [17] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

-
- [18] Z.-K. Huang. Bark classification using rbpnn based on both color and texture feature. *International Journal of Computer Science and Network Security*, 6(10):100–103, 2006.
- [19] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. Lopez, and J. V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification.
- [20] D. G. Lowe. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, 2:1150–1157, 1999.
- [21] J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281–297, 1967.
- [22] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.
- [23] W. S. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [24] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996.
- [25] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.
- [26] D. Park and P. Nikama. Lecture 11. Dostopno na: http://www.ics.uci.edu/~dramanan/teaching/ics273a_winter08/lectures/lecture11.pdf, 2013. Zadnji dostop: 5. 8. 2014.

-
- [27] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. Local binary patterns for still images. In *Computer Vision Using Local Binary Patterns*, volume 40 of *Computational Imaging and Vision*, pages 13–47. Springer London, 2011.
 - [28] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
 - [29] M. Sulc and J. Matas. Kernel-mapped histograms of multi-scale lbps for tree bark recognition. *28th International Conference on Image and Vision Computing New Zealand*, pages 82–87, 2013.
 - [30] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.
 - [31] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. Dostopno na: <http://www.vlfeat.org/>, 2008. Zadnji dostop: 6. 8. 2014.
 - [32] A. Vedaldi and B. Fulkerson. Vlfeat - an open and portable library of computer vision algorithms. Dostopno na: <http://www.robots.ox.ac.uk/~vedaldi/assets/pubs/vedaldi10vlfeat.pdf>, 2010. Zadnji dostop: 6. 8. 2014.
 - [33] M. Švab. Zbirka slik drevesnega lubja. Dostopno na: <http://go.vicos.si/trunk12>. Zadnji dostop: 8. 9. 2014.
 - [34] Z. Wang, B. Fan, and F. Wu. Local intensity order pattern for feature description. *International Conference on Computer vision*, pages 603–610, 2011.